Finite capacity planning algorithm for semiconductor industry considering lots priority

E.Mhiri*, M.Jacomino*, F.Mangione*
P.Vialletelle, G.Lepelletier****

**Univ.Grenoble Alpes, G-SCOP, F-38000 Grenoble,*
France (e-mails: Emna.Mhiri@grenoble-inp.fr; mireille.jacomino@grenoble-inp.fr; fabien.mangione@grenoble-inp.fr)
*** STMicroelectronics, F-38926 Crolles Cedex,*
France (e-mails: philippe.vialletelle@st.com ; guillaume.lepelletier@st.com)

Abstract: A finite capacity planning heuristic is developed for semiconductor manufacturing with high-mix low-volume production, complex processes, variable cycle times and reentrant flows characteristics. The proposed algorithm projects production lots trajectories (start and end dates) for the remaining process steps, estimates the expected load for all machines and balances the workload against bottleneck tools capacities. It takes into account lots' priorities, cycle time variability and equipment saturation. This algorithm helps plant management to define feasible target production plans. It is programmed in java, and tested on real data instances from STMicroelectronics Crolles300 production plant which allowed its assessment on the effectiveness and efficiency. The evaluation demonstrates that the proposed heuristic outperforms current practices for capacity planning and opens new perspectives for the production line management.

Keywords: Heuristic algorithm, finite capacity planning, semiconductor industry.

1. INTRODUCTION

Semiconductor manufacturing is a very complex process. It is composed of six major types of production operations as: oxidation and thermal treatment, film deposition, planarization, photolithography, etching and ion implantation. Figure 1 presents at simplified view the wafer fabrication process.

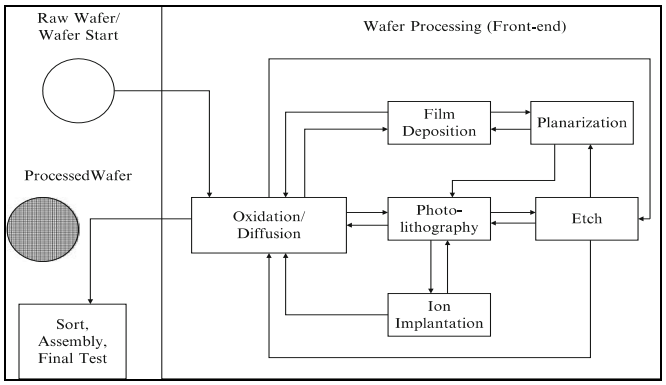

Fig. 1. Wafer fabrication Process. (Mönch et al., 2013)

The general purpose semiconductor manufacturers such as STMicroelectronics follows "make-to-order" business model, due to short product life cycles. The demand in this industry is characterized by diversity, both in terms of volume and production technologies. Besides this, reentrant production flows often result in huge cycle time variability. It requires, typically, 8 to 10 weeks to process a wafer with 300+ operations and 800+ elementary process steps (including metrology and cleaning steps) depending on the production technology (Shahzad et al., 2012). In semiconductor industry, each manufacturing process, named as process route, is divided into several operations where each operation comprises of multiple elementary steps with respective recipes. Different production equipment may be qualified for the same recipe and multiple recipes can be qualified on the same equipment. Identical equipment are also grouped into station-families that offer flexibility in production capacity requirements.

Therefore, production planning for semiconductor industry is very complex, especially in wafer processing phase (Chien et al., 2011). In this context, new methods and tools, leveraging Operational Research techniques and modern computation power have recently gained increasing attention. These tools are developed generally to minimize production costs (Catay et al., 2003), minimize total weighted lots tardiness (Habla et al., 2007), maximize profit (Ponsignon and Mönch, 2012) or maximize throughput (Chung and Jang, 2009) taking into account capacity constraints.

In most of proposed models, diversity in lots priorities and cycle time variability are not considered. These methods do not take into account due dates attached to the lots and also consider either fixed or average cycle time. However, in actual production lines, on-time delivery is highly important for customer satisfaction along with technology leadership, to gain market shares: the production plan has to integrate this aspect. Besides, as actual cycle time is widely spread and

skewed due to large variability in process steps, there is a significant difference between results obtained with averaged values and variable ones used in this paper.

In this paper, we focus on the production planning problem in wafer production lines and introduce a planning methodology that explicitly considers cycle time variability, lots priority and production capacity. The objectives are to minimize customer orders lateness and to optimize equipment utilization rate (to reduce cycle time variability). In this context, we propose a heuristic for capacity planning that pushes current Work In Progress (WIP) taking into account individual lots due dates, estimates expected equipment loads and balances workload and capacity of bottleneck equipment. Data used for model construction, performance evaluation and results validation is collected from STMicroelectronics Crolles300 wafer production line.

This paper is organized as follows. The next section presents a brief review on existing literature. Section 3 describes the proposed finite capacity planning algorithm followed by tests results in Section 4. Finally, Section 5 draws conclusions and future perspectives.

2. LITTERATURE REVIEW

In both academia and industry, there are numerous methods and tools used for capacity planning in the semiconductor manufacturing.

Besides traditional techniques like Manufacturing Resource Planning (MRP), Just In Time (JIT), Theory Of Constraints (TOC) and commonly used methods as spread sheets, linear programming and discrete-event simulation (Mönch et al., 2013), many researchers use heuristics or meta heuristics to resolve the problem of capacity planning. This is because of the complexity of the industrial context, higher-dimensional decision variables and higher required computational times. Chen et al. (2005) developed a Capacity Planning System (CPS) that determines lot's release time, fab starts plan, and the capability of the equipment for multiple semiconductor manufacturing fabs, considering a pull philosophy and an infinite equipment capacity. The effectiveness and efficiency of these systems are analysed using three performance indicators such as standard deviation in equipment utilization, number of oversaturated equipment and total extra capacity requirement exceeding equipment capacity limits. Milne et al. (2012) proposed an algorithm that blends linear programming with MRP heuristics for the IBM semiconductor fabrication facility in order to satisfy all demands on time. This algorithm generates a manufacturing release plan and work-in process priorities.

Besides the importance of these studies, they have major limitations as compared to our approach. They don't consider the finite capacity constraints. Furthermore, they use an estimated order's mean step waiting time and mean step cycle time as inputs to the proposed systems.

On the other hand, approximate methods have also been widely used to develop finite capacity planning systems for the semiconductor industry.

Rupp and Ristic (2000) presented a distributed finite capacity planning system using an iterative procedure based on the simulated annealing heuristic search algorithm to minimize the total production time of the set of orders. Horiguchi et al. (2001) proposed a simple finite capacity planning algorithm based on forward scheduling for WIP and backward scheduling for new orders. The objective of this algorithm is to calculate a release date for each order at each bottleneck position and to estimate its end date. In their study, authors have only considered the photolithography area at finite capacity and they don't take into account orders' due dates as well. Their approach is very similar to the capacitated MRP (MRP-C) algorithm of Tardif and Spearman (1997). Habla et al. (2007) suggested a production planning approach that takes into account finite capacity constraints with specific focus on bottleneck steps. They have formulated the problem into a mixed integer program (MIP) to determine completion time targets for bottleneck steps of lots. Lagrange relaxation and decomposition techniques are applied to solve the MIP approximately in a reasonable computational time. Chen et al. (2008) developed Finite Capacity Requirements Planning System (FCRPS) to balance the loading on various machines with same qualification and minimize mean absolute lateness of customer orders. This system, developed for multiple wafer fabs, considers orders due dates as well as equipment capacity, qualification and yield. It determines the order release time, start date, and equipment capability for each order. In this study, the step cycle time is estimated from the simulation of an AutoSched model.

In this paper, a heuristic algorithm is proposed for capacity planning that considers lots priorities, cycle time variability and capacity constraints.

3. FINITE CAPACITY PLANNING ALGORITHM

The goal of finite capacity planning algorithm is to calculate a planned start date for each individual lot in the WIP for all of its visits to a process step, and to estimate when it will be completed taking into account the lot's due date and stations families (i.e. groups of similar equipment) saturation. This algorithm consists of three main modules as–WIP projection module, workload accumulation and capacity analysis module and workload and capacity balancing module. As inputs, the developed system requires the horizon planning duration divided into weekly time buckets, lots due dates, the status of the WIP at the beginning of each projection period and the considered cycle time model. The following sections present each module in detail. The algorithm is executed by iterating it on time buckets of the planning horizon. Figure 2 depicts the flow of the developed system.

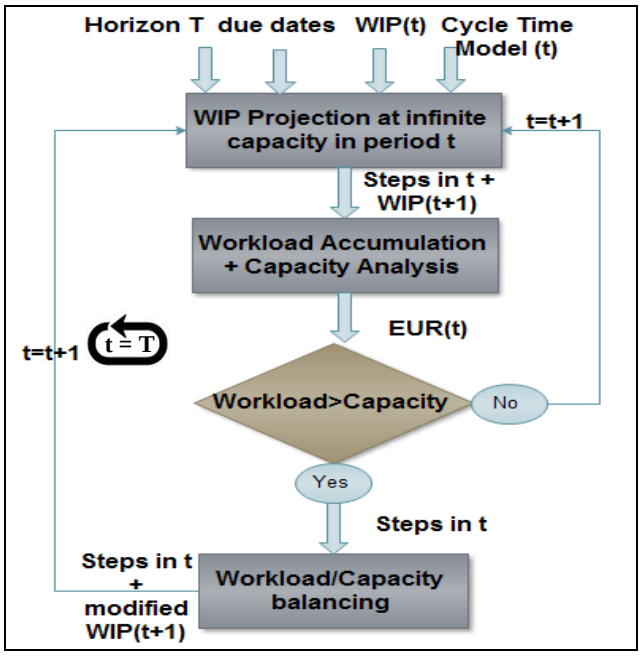

Fig. 2. Finite Capacity Planning Algorithm Flow.

3.1 WIP Projection Module

In a previous study (Mhiri et al., 2014), this module is explained in detail. WIP projection consists of translating the WIP inventory for each forward lot along its route from its current position during the considered period. Each lot has its own cycle-time model that computes process times for the individual steps based on necessary and sufficient speed for the remaining steps to achieve lots due dates and shared fine-tuned reference cycle time curves. The objective of this module is to estimate the periodic activity and future loading at station-families.

This module takes current WIP at lot level, lots due dates and a target cycle time model per step ($CTobj_{step}$) based on a semi-empirical formula as inputs. This formula multiplies the theoretical cycle time of the step ($CTTH_{step}$) that corresponds to the processing time, by a coefficient named $Xfactor_{step}$. $Xfactor_{step}$ depends on the theoretical cycle time of the step ($CTTH_{step}$), theoretical cycle time of route ($CTTH_{route}$) that corresponds to the sum of remaining steps process times, and objective cycle time of each route ($CTobj_{route}$) that takes into account queuing times (based on historical data), as presented below:

$$CTobj_{step} = CTTH_{step} \times Xfactor_{step} \quad (1)$$

$$Xfactor_{step} = \frac{CTTH_{route} \times \left(\frac{CTobj_{route}}{CTTH_{route}} - 1\right)}{\sum \sqrt{CTTH_{route} \times CTTH_{step}}} + 1 \quad (2)$$

This formula gives a rough estimation of queuing time at each step. The principle of projection consists of computing the objective cycle time for each step of each route, according to the above formula. A penalty is also added to $Xfactor_{step}$ for bottleneck steps to take into account the saturation of the toolset.

Then, from its current position in its route, a coefficient named $Xfactor_{lot}$ is computed for each lot. The $Xfactor_{lot}$ corresponds to the ratio between the remaining time to reach the ship date and the remaining objective cycle time of the remaining steps. The ship date is equal to the maximum between the due date as defined by the customer and the minimum feasible ship date that is equal to the sum of WIP extraction date and process times of the remaining steps. Afterwards, the steps are projected according to the cycle time that is equal to $CTobj_{step} \times Xfactor_{lot}$.

Finally, we compute the number of tracks per period (*TrackIn* for wafers entering a step, *TrackOut* for those completing it) and quantity of WIP at the beginning and end of each period.

To further explain the concept of WIP projection, a simple instance is tested with input data inspired from the real data provided by STMicroelectronics Crolles 300 production line. The considered WIP is composed of 10 lots with different due dates. The Table 1 presents, for each lot, the number of remaining steps, remaining time to meet due date from WIP extraction date, remaining objective cycle time and Xfactor.

Table 1. WIP data

Lot	Number of remaining steps	Remaining time to due date in days	Remaining objective cycle time in days	$Xfactor_{lot}$
Lot 1	6	9,42	4,35	2,16
Lot 2	4	0,42	3,71	0,206
Lot 3	2	2,42	2,29	1,06
Lot 4	8	0,42	6,79	0,249
Lot 5	6	2,42	3,71	0,404
Lot 6	4	9,42	6,83	2,542
Lot 7	8	2,42	3,875	0,354
Lot 8	4	0,42	4,33	0,206
Lot 9	4	2,42	3,71	0,652
Lot 10	6	2,42	4,33	0,558

Figure 3 illustrates projection results of the 10 lots during the first period of the planning horizon. It shows the start and end dates for each remaining step in the WIP, queue time and processing time for each step during considered period. There are some steps which start in the first period and finish in the subsequent periods of the planning horizon. Figure 3 demonstrates that the projection engine allows the extension of steps queuing times, when we are far from the due date and it shrinks steps cycle times in the case of reduced margin

between WIP extraction date and due dates. Lot2, lot4 and lot 8 are not delivered on time. Their shipping date is equal to the sum of WIP extraction date and remaining process times.

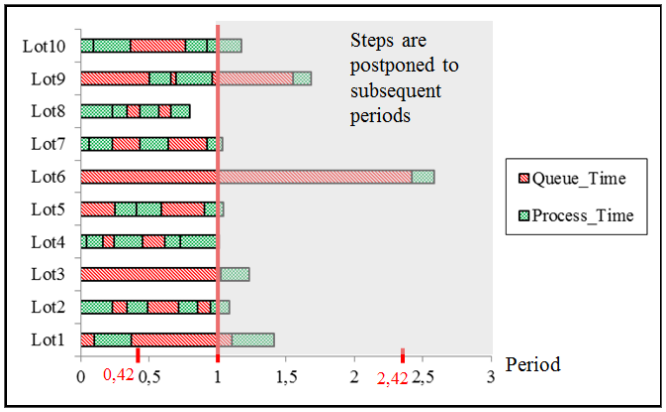

Fig. 3. WIP projection results for the first period of the planning horizon.

3.2 Workload Accumulation and Capacity Analysis Module

After WIP projection, the equipment loading, over each period t, is computed with an existing tool, named CAPACE at STMicroelectronics, based on the assumption of infinite station-family capacities.

The inputs for the engine are:

- The number of *TrackIn* over period t,
- The model for station-families, i.e. the number of tools in station family, maximum tolerable loading (*EUR_MAX*), availability by period (*stnfam_availTimePerPeriod*) and batch load (*BatchLoad*) which is the percentage of time to load a batch composed of several lots,
- The recipe model which corresponds to the qualified station-families for each recipe with its matching processing time.

To optimize the computation time, station-families are distributed in balancing groups. This approach enables to decompose the problem into small sub-problems. A balancing group is a set of station-families that have same qualifications and shares same recipes. Then, the workload of each station family is computed in two steps as under:

Step1: Compute the total time consumed to process a wafer (*CumulConsoTimeWafer*) as sum of the product of number of Track In at each recipe ($TrackIn_{recipe}$) that is processed at the station family by its consumed time to process a wafer ($ConsoTimeWafer_{recipe,stnfam}$):

$$CumulConsoTimeWafer = \sum_{recipe} TrackIn_{recipe} \times ConsoTimeWafer_{recipe,stnfam} \quad (3)$$

Step2: Compute Equipment Utilization Rate (*EUR*) of each station family. It is equal to the ratio of total time consumed to process a wafer (*CumulConsoTimeWafer*) by availability percentage per period (*stnfam_availTimePer Period*), batch load (*BatchLoad*) and capacity (*EUR_MAX*).

$$EUR = \frac{CumulConsoTimeWafer}{stnfam\_availTimePerPeriod \times BatchLoad \times EUR\_MAX} \quad (4)$$

Afterwards, the system uses a linear program to optimize workload balancing of stations families, belonging to the same balancing group.

So, the output of this system is the Equipment Utilization Rate (*EUR*) at each station family over a period t.

For the example cited above, we consider that the remaining steps of the 10 lots (Stepi.j, step j of i^{th} lot, i= {1...10}, j= {1...8}) are processed in 6 station families {*Sf1*, *Sf2*, *Sf3*, *Sf4*, *Sf5*, *Sf6*}. Figure 4 illustrates the saturation at each station family (*EUR/EUR_MAX)* during the first period of the planning horizon.

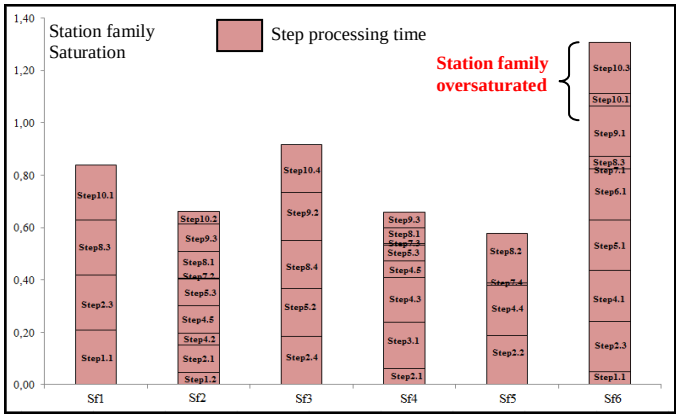

Fig. 4. Workload accumulation results for the first period of the planning horizon.

Figure 4 shows that the station family *Sf6* is oversaturated. Its workload exceeds its load threshold (*EUR_MAX*).

3.3 Workload/Capacity Balancing Module

As a result of workload accumulation module, we may find that the workload of some stations families exceeds maximal capacity. In this case, the station family is unable to process all its affected steps during a considered period so the balancing module is needed. The goal of this module is to postpone supplementary lots in order to bring back workload of oversaturated station-families to their maximal saturation.

The algorithm for workload/capacity balancing module is as follows:

1. Sort oversaturated station families in descending order of workload.

2. Select lots executed on the most loaded station family having *EUR (t) >EUR_MAX*.

3. Sort selected lots in descending order of $XFactor_{lot}$ (lot due date-current date > Remaining objective cycle time of remaining steps) during the period.

4. For the first selected lot in the sorted list, beginning with the step executed in considered oversaturated station family,

shift latest projected steps from the end of the period to the beginning of the next period.

5. For each shifted step, the processing time is removed from the process time of each station-families of the set of its qualified processing station families while considering the percentage of *TrackIn* in each one. Indeed, the removed process time is equal to the product of the time consumed of the shifted recipe by the number of wafers of the shifted lot (*WIP_Quantity*) and the percentage of *TrackIn* of the shifted recipe for the considered station family ($TrackIn_{recipe,stnfam}$) compared to its total number of track in ($TotalTrackIn_{recipe}$).

$$Removed\ Process\ Time = ConsoTimeWafer_{recipe,stnfam} \times WIP\_Quantity \times \frac{TrackIn_{recipe,stnfam}}{TotalTrackIn_{recipe}} \quad (5)$$

6. Remove the treated oversaturated station family from the initial list of the oversaturated station families.

7. Repeat step1 for the refreshed list of station-families.

8. Repeat steps 2, 3,4,5,6 and 7 for all lots and all stations families until the workload/capacity balance is achieved for all stations family over the period t.

Hence, this module modifies steps projection over period t and the WIP for the beginning of the next period t+1.

For instance, to balance the capacity and the workload of the station family *Sf6* in the considered example, the Balancing Module selects lots 1, 2, 4, 5, 6, 7, 8, 9 and 10 executed on this resource (Figure 4). These lots are classified in descending order of $Xfactor_{lot}$ as the following: lot6, lot1, lot9, lot10, lot5, lot7, lot4, lot8 then lot 2. So, we begin by shifting step 6.1 of lot 6 (composed of 25 wafers) to the next period of the planning horizon. The loading of *Sf6* decreases (0,704-0,12=0,584<0,62). Therefore, saturation of *Sf6* becomes less than its maximum capacity.

4. RESULTS AND DISCUSSION

The finite capacity planning algorithm was implemented with JAVA programming language. The experiments were run with an Intel Core i5, 2.7GHz, 4.0 GB RAM.

In the previous section, we test a simple instance during the first period of the planning horizon to explain the principle of each module of the proposed finite capacity planning algorithm. This instance is tested throughout a planning horizon divided into five daily time buckets. The execution time of this instance is about 2 seconds. The final schedule for this instance during the planning horizon is illustrated in figure 5. This figure shows an extension of the queue time for lot1 and lot 6 having a large margin to reach their due dates. In addition to step 6.1 which is shifted to period 2 as it is explained above, step 9.2 of lot 2 is shifted to the third period because of the oversaturation of station family *Sf5* processing this step in period 2 and because lot 9 has the most important Xfactor in this period.

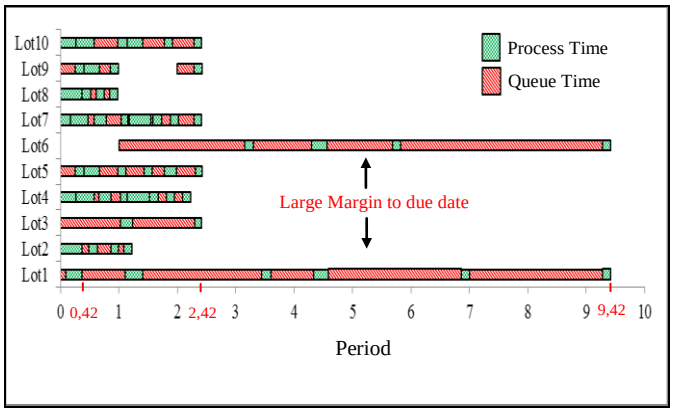

Fig.5. An example of a schedule established by the finite capacity planning system.

The results indicate that there are only 3 lots (lot2, lot4 and lot8) that aren't delivered on time with a delay of 16 hours, 24 hours and 9 hours, respectively. Besides, in the final production planning, there is no station family that its saturation exceeds its maximum capacity.

To evaluate the ability of the proposed approach to tackle the real world problems, a real instance designed with a realistic size and complexity is tested. The real instance provided by STMicroelectronics corresponds to a WIP composed of several thousand lots with 200 to 300 remaining steps for each one. These steps are processed with several hundred station-families. As planning horizon, we consider one month divided into 4 time buckets (weeks). The execution time of this instance is less than 5 minutes. In the production schedule established by the developed system, 99% of projected lots are delivered on time. Furthermore, there is no station family where saturation exceeds its maximum capacity and maximum saturation per period is about 83%. The obtained results show that the implementation of the finite capacity planning system in real Fabs seems very interesting to minimize lots lateness and optimize equipment utilization rate.

5. CONCLUSIONS AND PERSPECTIVES

In this paper, we proposed a decision support tool for finite capacity planning in semiconductor wafer production lines. Compared to the related literature presented in Section 2, the proposed approach takes into account capacity constraints, lots priorities and cycle time variability. It generates the start and end dates for each lot's step as well as the estimated balanced loading by time bucket for each station family.

The results of some preliminary computational experiments show that the number of delayed lots could be minimized and the average equipment utilization rate could be optimized significantly by using the developed system. Besides, the test of this system for a real instance is achieved in less than 5 minutes of computation time which seems to be sufficient for planning problems with a horizon of weeks up to months in real time situations.

There are several directions for future research. First, more computational experiments are necessary. Second, to enhance the accuracy of the developed system, it seems interesting to add a cycle time estimation module in the end of the algorithm. This module, based on the queuing theory (Leachman, 2012), computes an estimated cycle time that takes into account the process mix and saturation of the station-families. Third, it has to be investigated if the computation times can be further reduced by the use of the parallel programming. Fourth, to evaluate the performance of this system, it seems interesting to compare the test results of real instances using the developed system with those obtained with the existent capacity planning tool being used by STMicroelectronics.

REFERENCES

Catay, B., Erenguc, S.S., and Vakharia, A.J. (2003). Tool capacity planning in semiconductor manufacturing. *Computers & Operations Research*, 30(9), 1349–1366.

Chen, J. C., Chen, C. W., Lin, C. J., and Rau. H. (2005). Capacity planning with capability for multiple semiconductor manufacturing fabs. *Computers and Industrial Engineering*, 48(4), 709-732.

Chen, C.W., Chen, J.C., and Lin, C.J. (2008). Finite capacity requirements planning with equipment capability and dedication for semiconductor manufacturing. In *Proceedings of the 9th Asia Pacific Industrial Engineering & Management Systems Conference (APIEMS)*, 1310-1319.

Chien, C-F., Dauzère-Pérès, S., Ehm, H., Fowler, J.W., Jiang Z., Krishnaswamy, S., Mönch, L., and Uzsoy, R. (2011). Modeling and analysis of semiconductor manufacturing in a shrinking world: challenges and successes. *European Journal of Industrial Engineering*, 5(3), 254–271.

Chung, J., and Jang, J. (2009). A WIP Balancing Procedure for Throughput Maximization in Semiconductor Fabrication. *IEEE Transactions on semiconductor manufacturing*, 22(3), 381 – 390.

Habla, C., Mönch, L., and Drießel, R. (2007). A Finite Capacity Production Planning Approach for Semiconductor Manufacturing. In *Proceedings of the 3rd Annual IEEE Conference on Automation Science and Engineering*, 82-87, Scottsdale, AZ, USA.

Horiguchi, K., Raghavan, N., Uzsoy, R., and Venkateswaran, S. (2001). Finite-capacity production planning algorithms for a semiconductor wafer fabrication facility. *International Journal of Production Research*, 39 (5), 825-842.

Leachman, R. C. (2012). The Engineering Management of Speed. In *Proceedings of the 2012 Industry Studies Association Annual Conference.* Pittsburgh.

Mhiri,E., Jacomino, M., Mangione, F., Vialletelle, P. and Lepelletier, G. (2014). A step toward capacity planning at finite capacity in semiconductor manufacturing. *Winter Simulation Conference* , Savannah, Georgia.

Milne, R. J., Wang, C-T., Yen, C-K.A., and Fordyce, K. (2012). Optimized material requirements planning for semiconductor manufacturing, *Journal of the Operational Research Society*, 63(11), 1566–1577.

Mönch, L., Fowler, J.W., and Mason, S.J. (2013). *Production Planning and Control for Semiconductor Wafer Fabrication Facilities: Modeling, Analysis, and Systems*, 207-246.Springer, New York.

Ponsignon, T., Mönch, L. (2012). Heuristic approaches for master planning in semiconductor manufacturing. *Computer Operations Research*, 39(3), 479–491.

Rupp, T. M., and Ristic, M. (2000). Fine planning for supply chains in semiconductor manufacture. *Journal of Materials Processing Technology*, 107(2000),390-397.

Shahzad, M.K., Chaillou,T., Hubac,S., Siadat,A., and Tollenaere, M. (2012). A yield aware sampling strategy for tool capacity optimization. *International Conference on Artificial Intelligence (ICAI)*, Las Vegas, United States.

Tardif, V., and Spearman, M. L. (1997). Diagnostic scheduling in a finite-capacity environment. *Computers and Industrial Engineering*, 32 (4), 867–878.